

Migration von Daten nach Nuclos

- [Migration mit Bordmitteln](#)
- [Migration mit SQL](#)
 - [Businessobjekte ohne Statusmodell](#)
 - [Codebeispiele zum Verwenden von IDFactories:](#)
 - [Businessobjekte mit Statusmodell](#)

Migration mit Bordmitteln

Für die Migration mit Bordmitteln stehen Ihnen die Importstrukturdefinitionen zur Verfügung, siehe auch [Import & Export](#).

Für komplexe Migrationen, die mit nennenswerten Änderungen von Datenstrukturen und Umformatierungen o.ä. einhergehen, sind diese Mechanismen nur bedingt geeignet. Zum einen sind in der Regel manuelle Zwischenschritte erforderlich (z.B. Export der Daten aus dem Altsystem nach CSV, Löschen bereits migrierter Daten vor einer erneuten Migrationsdurchführung, Sicherstellung gleicher Formatierungen und Zeichensätze, etc.), zum anderen leidet schnell die Übersichtlichkeit, auch bzgl. einer Ausführungsreihenfolge, bei hoher Anzahl von Importstrukturdefinitionen. Auch ist eine auswertbare Protokollierung und / oder Fehlerbehandlung nur bedingt möglich. Nicht zuletzt ist eine Umformatierung von Daten unter Zuhilfenahme weiterer Datenquellen mit diesen Werkzeugen nicht möglich.

Migration mit SQL

Für komplexe Migrationen empfehlen wir die Erstellung eines SQL Skriptes (wenn die Datenbank des Altsystems auf dem gleichen DBMS beruht wie die Nuclos Datenbank) oder eines Programmes (wenn unterschiedliche DBMS), das z.B. mittels JDBS Datenbankverbindungen zu beiden Datenbanken herstellt und die Migration synchron und vollautomatisch durchführt. Auch erleichtert dies die Herstellung einer Idempotenz der Ausführung, d.h. ein erneutes Migrieren auf Knopfdruck mit selbständigem Löschen aller schon bestehenden Inhalte, zur jederzeitigen wiederholten Migration nach Änderungen der Datenstruktur oder nach Korrekturen im Fehlerfall.



Für eine Migration mittels SQL empfiehlt es sich, vor Ausführung sämtliche Constraints zu deaktivieren, und am Ende der Migration wieder zu aktivieren, um die Reihenfolge der zu migrierenden Tabellen während der Migrationsausführung frei wählen zu können (und auf eine Wartbarkeit des Migrationsskriptes hin optimieren zu können). In den meisten DBMS stehen Systemtabellen zur Verfügung, aus denen man alle Constraints dynamisch ermitteln kann (z.B. Tabelle user_constraints in Oracle).



Für eine idempotente Migration sollte das Skript im ersten Schritt sämtliche Inhalte der Zieltabellen der Migration löschen, auf diese Weise stellt das Skript selbst einen immer gleichen (leeren) Ausgangszustand für die Migration sicher.

Bei der Migration muss grundsätzlich unterschieden werden zwischen Businessobjekten ohne Statusmodell und Businessobjekten mit Statusmodell (siehe [Statusmodell](#)). Bei den im Folgenden beschriebenen Migrationsanforderungen handelt es sich um Mindestanforderungen. Situationsbedingt können im Einzelfall weitere Anforderungen hinzukommen.

Businessobjekte ohne Statusmodell

Bei der Befüllung von Tabellen ohne Statusmodell ist lediglich zu beachten, dass die Systemattribute INTID, DATCREATED, STRCREATED, DATCHANGED, STRCHANGED, INTVERSION korrekt und sinnvoll befüllt werden.

Systemattribut	Hinweise
INTID	<p>Empfohlen wird die Befüllung über die Nuclos Sequenz IDFACTORY.</p> <p>Euch möglich ist eine Befüllung mit beliebiger Zahl, es ist dabei die Eindeutigkeit innerhalb der Tabelle sicherzustellen.</p> <p>Um Verletzungen der Eindeutigkeit nach der Migration zu vermeiden, ist die Sequenz IDFACTORY auf den über alle Tabellen vergebenen höchsten Wert +1 zu setzen.</p>
DATCREATED	Beliebiges Datum, z.B. das Datum der ursprünglichen Erstellung des Datensatzes im Altsystem oder das aktuelle Datum.
STRCREATED	Beliebiger String, z.B. der Benutzername der ursprünglichen Erstellung des Datensatzes im Altsystem oder ein generischer String wie "MIGRATION".
DATCHANGED	Beliebiges Datum, z.B. das Datum der letzten Änderung des Datensatzes im Altsystem oder das aktuelle Datum.
STRCHANGED	Beliebiger String, z.B. der Benutzername der letzten Änderung des Datensatzes im Altsystem oder ein generischer String wie "MIGRATION".
INTVERSION	1

Codebeispiele zum Verwenden von IDFactories:

1. Postgres

```
INSERT INTO T_EO_KUNDE VALUES (NEXTVAL('idfactory'), ...)
```

2. Oracle

```
INSERT INTO T_EO_KUNDE VALUES (IDFACTORY.nextval, ...)
```

Businessobjekte mit Statusmodell

Zusätzlich zu den Systemattributen, wie sie auch für Businessobjekte ohne Statusmodell (siehe vorhergehender Abschnitt) zu migrieren sind, sind weitere Systemattribute zu befüllen.

Systemattribut	Hinweise
STRUID_NUCLOSSTATE	Referenz auf Status (siehe Statusmodell).
STRUID_NUCLOSPROCESS	Referenz auf Aktion (siehe Aktionen), soweit für das betreffende Businessobjekt Aktionen definiert sind.
STRNUCLOSSYSTEMID	Beliebiger String, der den Datensatz fachlich eindeutig identifiziert.
BLNNUCLOSDELETED	false
STRNUCLOSORIGIN	NULL

Darüberhinaus sind folgende Systemtabellen zu befüllen, dabei ist jeweils ein Datensatz je migriertem Datensatz für Businessobjekte mit Statusmodell einzutragen.

Systemtabelle	Attribut	Hinweise
T_UD_GENERIC OBJECT	INTID	<Businessobjekt>.INTID, siehe oben (identisch zu befüllen)
	DATCREATED	<Businessobjekt>.DATCREATED, siehe oben (identisch zu befüllen)
	STRCREATED	<Businessobjekt>.STRCREATED, siehe oben (identisch zu befüllen)
	DATCHANGED	<Businessobjekt>.DATCHANGED, siehe oben (identisch zu befüllen)
	STRCHANGED	<Businessobjekt>.STRCHANGED, siehe oben (identisch zu befüllen)
	INTVERSION	1
	STRUID_T_MD_M ODULE	Referenz auf T_MD_ENTITY (STRUID), das das zugehörige Businessobjekt repräsentiert.
T_UD_GO_STA TEHISTORY	INTID	Empfohlen wird die Befüllung über die Nuclos Sequenz IDFACTORY. Euch möglich ist eine Befüllung mit beliebiger Zahl, es ist dabei die Eindeutigkeit innerhalb der Tabelle sicherzustellen. Um Verletzungen der Eindeutigkeit nach der Migration zu vermeiden, ist die Sequenz IDFACTORY auf den über alle Tabellen vergebenen höchsten Wert +1 zu setzen.
	DATCREATED	<Businessobjekt>.DATCREATED, siehe oben (identisch zu befüllen)
	STRCREATED	<Businessobjekt>.STRCREATED, siehe oben (identisch zu befüllen)
	DATCHANGED	<Businessobjekt>.DATCHANGED, siehe oben (identisch zu befüllen)
	STRCHANGED	<Businessobjekt>.STRCHANGED, siehe oben (identisch zu befüllen)
	INTVERSION	1
	STRUID_T_MD_ST ATE	<Businessobjekt>.STRUID_NUCLOSSTATE, siehe oben (identisch zu befüllen)
	INTID_T_UD_GEN ERICOBJECT	T_UD_GENERICOBJECT.INTID bzw. <Businessobjekt>.INTID